Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Applied Numerical Mathematics 49 (2004) 39–61
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

# Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization

E. Kokiopoulou [a,*], C. Bekas [a,1], E. Gallopoulos [b]

[a] *Computer Science and Engineering Department, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455, USA*

[b] *Computer Engineering and Informatics Department, University of Patras, 26500 Patras, Greece*

**Abstract**

A matrix-free algorithm, IRLANB, for the efficient computation of the smallest singular triplets of large and possibly sparse matrices is described. Key characteristics of the approach are its use of Lanczos bidiagonalization, implicit restarting, and harmonic Ritz values. The algorithm also uses a deflation strategy that can be applied directly on Lanczos bidiagonalization. A refinement postprocessing phase is applied to the converged singular vectors. The computational costs of the above techniques are kept small as they make direct use of the bidiagonal form obtained in the course of the Lanczos factorization. Several numerical experiments with the method are presented that illustrate its effectiveness and indicate that it performs well compared to existing codes.


*Keywords:* Lanczos bidiagonalization; Implicit restarting; Harmonic Ritz values; Deflation; Refined singular vectors; Pseudospectrum

## 1. Introduction

Consider the singular value decomposition (SVD) $A = U\Sigma V^*$ of a matrix $A \in \mathbb{C}^{m\times n}$, where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ and, without loss of generality, $n \geqslant m$. Denote its singular triplets by $(\sigma_i, u_i, v_i)$, $i = 1, \ldots, \min(n, m) \equiv m$, where $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > \sigma_{r+1} = \cdots = \sigma_m = 0$. In this paper we are interested in computing few of the smallest singular triplets of a general large sparse matrix. This problem arises

* Corresponding author.
*E-mail addresses:* kokiopou@cs.umn.edu (E. Kokiopoulou), bekas@cs.umn.edu (C. Bekas), stratis@ceid.upatras.gr (E. Gallopoulos).

[1] Authors supported by graduate fellowships of the Hellenic Bodossaki Foundation. This work was completed while the authors were with the Computer Engineering and Informatics Department, University of Patras, Greece.

0168-9274/$30.00 © 2003 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2003.11.011

in several important applications including total least squares [6], information retrieval [9], image and signal processing [50], control [13] and matrix pseudospectra [49].

The computation of few extremal singular triplets of large sparse matrices has been the focus of many research efforts, see [3,4,11,16,29,32,42,45,47] as well as [2,12,17,23,41,48] and numerous references therein. Recent needs in applications such as the ones mentioned earlier, however, have motivated research oriented towards the development of algorithms for the computation of the smallest singular triplets, a problem that is acknowledged to challenge the capabilities of current state-of-the-art software, e.g., see [1,8,14,18,20,21,34].

It is common practice to approximate singular values by computing the eigenvalues of related Hermitian eigenproblems. Furthermore, since computing the smallest eigenvalues of a matrix is equivalent to computing the largest eigenvalues of its inverse, significant work has been done on "shift-and-invert" techniques. For example, this approach was adopted in the MATLAB (version 6) svds routine, that is based on ARPACK [32]; the latter, implements one of the most successful theoretical frameworks for the effective implicitly restarted Arnoldi technique, based on seminal work of Sorensen, Lehoucq and collaborators. However, as the size of the matrices increases, this approach becomes too expensive in terms of storage and computational costs, as it requires the factorization of and solution with large sparse, possibly indefinite matrices. Developments that attempt to remedy this problem concern inexact inverse iteration and inexact inverse Lanczos methods (see, for example, [27] and [2, Section 11.2]). An alternative approach that avoids such solves and is frequently effective is based on the use of harmonic Ritz values [38,45].

In this paper we propose and investigate an algorithm, we call IRLANB, that is based on Lanczos bidiagonalization (LBD), a method for computing singular values originally due to Golub and Kahan [15]. This is a matrix-free method for the computation of the singular triplets, thus the only operations with $A$ are matrix vector multiplications with it and its Hermitian adjoint $A^*$. We enhance the LBD algorithm with state-of-the-art technology for the effective computation of few small singular triplets of large and possibly sparse matrices. These improvements are described in the paper, whose structure is as follows. In Section 2 we review Lanczos bidiagonalization and describe its limitations when deployed to compute the smallest singular triplets. In Section 3 we show how to incorporate implicit restarts, introduced in [47], that permit Lanczos bidiagonalization to maintain limited storage and computational requirements per restart. In Section 4 we study the use of Ritz and harmonic Ritz values as implicit shifts. In Section 5 we show how to apply the orthogonal deflation transformation proposed in [46] in the context of Lanczos bidiagonalization to also make it more effective when the singular values are clustered. In Section 6 we show how to use refinement, originally proposed for eigenvectors in [24], to enhance the computation of singular triplets. In Section 7, we describe the overall structure of IRLANB. Finally, in Section 8 we describe numerical experiments that illustrate the behavior of IRLANB in various cases and compare its performance with related methods.

Implicit restarting in the context of LBD was first studied by Björck et al. in [5] and later Larsen combined it with partial reorthogonalization in [28,31]. After submitting the first version of this paper, we became aware of a contribution conducted independently by Jia and Niu [25] which proposes implicitly restarted LBD using "refined shifts" [25], in order to compute a few largest or smallest singular values. An important difference between IRLANB and the above approaches is the use of harmonic Ritz values in order to effectively approximate the smallest singular values of the matrix. Furthermore, in IRLANB we have adopted a philosophy that acknowledges the inherent difficulties of the problem and attempts to address them by combining state-of-the-art techniques, such as deflation and use of refined residuals.

### 1.1. Definitions and equivalent Hermitian eigenproblems

The following well-known connections (see, e.g., [17, Section 8.6]) between the SVD and the eigendecompositions of the following Hermitian matrices

$$A^*A, \quad AA^* \quad \text{and} \quad C = \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix} \in \mathbb{C}^{(m+n)\times(m+n)},$$

are fundamental to our discussion:

$$V^*\big(A^*A\big)V = \operatorname{diag}\Big(\sigma_1^2, \ldots, \sigma_m^2, \underbrace{0, \ldots, 0}_{n-m}\Big), \tag{1}$$

$$U^*\big(AA^*\big)U = \operatorname{diag}\big(\sigma_1^2, \ldots, \sigma_m^2\big). \tag{2}$$

Partitioning $V$ as $V = [V_1, V_2]$, where $V_1$ consists of $m$ columns and setting

$$Y = \frac{1}{\sqrt{2}} \begin{bmatrix} -U & 0 & U \\ V_1 & \sqrt{2}V_2 & V_1 \end{bmatrix},$$

then, $Y$ is an orthonormal eigenbasis for the augmented matrix $C$ and

$$Y^*CY = \operatorname{diag}\Big(-\sigma_1, \ldots, -\sigma_m, \underbrace{0, \ldots, 0}_{n-m}, \sigma_1, \ldots, \sigma_m\Big). \tag{3}$$

The above equivalences provide a convenient framework when seeking few singular values of large matrices because they permit the computation of singular triplets using Hermitian eigensolvers as a black box. The problem has been studied in the literature and there exist several software packages for its solution (see, e.g., [28,32,52]), and software based on the Jacobi–Davidson method (e.g., see, [22] and Section 8). Nevertheless, when seeking few small singular triplets, as we do in this paper, several complications arise that must be addressed [40,45].

In particular, since we are interested in the smallest singular values of $A$, equivalent targets are the smallest eigenvalues of either $AA^*$ or $A^*A$, or interior eigenvalues of $C$ (in the latter two, excluding spurious zeroes). Observe that, while squaring the singular values of $A$ will induce an increase of the separation of the largest ones, it will also cause a corresponding clustering of the smallest ones; this can cause problems for Hermitian eigensolvers [39, Section 11.7]. Furthermore, if $A$ is ill-conditioned, and we denote by $\kappa(A)$ its condition number with respect to the 2-norm, the squaring of the condition number, $\kappa(A^*A) = \kappa(AA^*) = \kappa(A)^2$, is likely to cause significant loss of accuracy for small singular values. Note that for rectangular matrices the above analysis holds if we refer instead to the "effective condition" $\|A\|_2\|A^\dagger\|_2$, where $A^\dagger$ denotes the pseudoinverse of $A$ (see [7, p. 28]). If, on the basis of relation (3), we select instead to recover the singular triplets of $A$ from the eigenvalues of the augmented matrix $C$, we have to approximate interior eigenvalues. Unfortunately, such a computation also challenges the performance of Hermitian eigensolvers, e.g., their convergence behavior becomes irregular (see, e.g., [45, Section 5]). Furthermore, since each singular value corresponds to an eigenvalue pair, $\pm\sigma_i$, Hermitian eigensolvers tend to take twice the number of iterations. An additional difficulty stems from the increased length $(m + n)$ of the basis vectors and corresponding increase in the storage requirements, from which approximations to the singular values are drawn.

## 2. Lanczos bidiagonalization

We next describe Lanczos bidiagonalization (LBD) that holds a central role in our framework. LBD was originally proposed by Golub and Kahan (cf. [15] and [17, Section 9.3.3]) as a process for transforming a matrix $A \in \mathbb{C}^{m\times n}$ to upper bidiagonal form, $B \in \mathbb{R}^{m\times n}$. In line with the bidiagonalization algorithms presented elsewhere in the literature, we will consider a version of the process that transforms $A$ to lower bidiagonal form (first discussed in [37]). In fact, our discussion owes a lot to the work of Larsen in [29]. After $k < m$ (successful) steps, LBD produces two blocks of Lanczos vectors

$$U_{k+1} = [u_1, u_2, \ldots, u_{k+1}] \in \mathbb{C}^{m\times(k+1)}, \qquad V_k = [v_1, v_2, \ldots, v_k] \in \mathbb{C}^{n\times k},$$

whose columns are orthonormal bases for the Krylov subspaces $\mathcal{K}_{k+1}(AA^*, u_1)$, $\mathcal{K}_k(A^*A, v_1)$, respectively (where, as usual, for any square matrix $G \in \mathbb{C}^{n\times n}$, $\mathcal{K}_m(G, r) \equiv \text{span}\{r, Gr, \ldots, G^{m-1}r\}$) and satisfy the following relations:

$$AV_k = U_{k+1}B_k, \tag{4}$$

$$A^*U_{k+1} = V_kB_k^* + \alpha_{k+1}v_{k+1}e_{k+1}^*, \tag{5}$$

where the matrix $B_k \in \mathbb{R}^{(k+1)\times k}$ has real elements and is lower bidiagonal:

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}. \tag{6}$$

The outline of LBD is provided in Algorithm 1.

**Algorithm 1** (*Lanczos bidiagonalization*) (*cf.* [2,15,29]). The real scalars $\alpha_i, \beta_i$ are the diagonal and subdiagonal elements respectively, of the bidiagonal matrix $B_k$.

*Input*: $A \in \mathbb{C}^{m\times n}$, starting vector $p_0 \in \mathbb{C}^m$ and scalar $k$
*Output*: Bidiagonal matrix $B_k \in \mathbb{R}^{(k+1)\times k}$ and orthogonal bases
$U_{k+1} \in \mathbb{C}^{m\times(k+1)}$, $V_k \in \mathbb{C}^{n\times k}$

1. Set $\beta_1 = \|p_0\|_2$, $u_1 = p_0/\beta_1$ and $v_0 = 0$
2. **for** $i = 1, 2, \ldots, k$
3. $r_i = A^*u_i - \beta_i v_{i-1}$
4. $\alpha_i = \|r_i\|_2$
5. $v_i = r_i/\alpha_i$
6. $p_i = Av_i - \alpha_i u_i$
7. $\beta_{i+1} = \|p_i\|_2$
8. $u_{i+1} = p_i/\beta_{i+1}$
9. **end**

Following the execution of LBD, the singular values of $B_k$ could be used as approximations to the singular values of $A$. If we premultiply both sides of (5) with $A$ and use (4) we obtain

$$AA^*U_{k+1} = (AV_k)B_k^* + \alpha_{k+1}Av_{k+1}e_{k+1}^* = U_{k+1}B_kB_k^* + \alpha_{k+1}Av_{k+1}e_{k+1}^*. \tag{7}$$

However, from the LBD algorithm (cf. lines 6–8 of Algorithm 1) we can also write

$$\begin{aligned} AA^*U_{k+1} &= U_{k+1}B_kB_k^* + \alpha_{k+1}(\alpha_{k+1}u_{k+1} + \beta_{k+2}u_{k+2})e_{k+1}^* \\ &= U_{k+1}\left(B_kB_k^* + \alpha_{k+1}^2 e_{k+1}e_{k+1}^*\right) + \alpha_{k+1}\beta_{k+2}u_{k+2}e_{k+1}^*. \end{aligned} \tag{8}$$

Matrix $B_kB_k^* + \alpha_{k+1}^2 e_{k+1}e_{k+1}^*$ is real symmetric and tridiagonal, therefore, in exact arithmetic, relation (8) is a symmetric Lanczos factorization and hence LBD is equivalent to symmetric Lanczos iteration on $AA^*$.

It is also known that there is an equivalence between LBD applied on $A$ and Lanczos applied on the augmented matrix $C$ [17, Section 9.3.2]. In particular, consider the starting vector

$$q_1 = \Big(u_1^*, \underbrace{0, \ldots, 0}_{n}\Big)^*, \quad \|u_1\|_2 = 1.$$

After $2k$ steps of Lanczos with starting vector $q_1$ the following relation holds:

$$CQ_{2k} = Q_{2k}T_{2k} + \beta_{k+1}q_{2k+1}e_{2k}^* = Q_{2k}T_{2k} + \beta_{k+1}\begin{pmatrix} u_{k+1} \\ 0 \end{pmatrix}e_{2k}^*, \tag{9}$$

where $q_{2j-1} = (u_j^*, 0)^*$ and $q_{2j} = (0, v_j^*)^*$, $j = 1, \ldots, 2k$ and

$$T_{2k} = \begin{pmatrix} 0 & \alpha_1 & & & \\ \alpha_1 & 0 & \beta_2 & & \\ & \beta_2 & 0 & \ddots & \\ & & \ddots & \ddots & \alpha_k \\ & & & \alpha_k & 0 \end{pmatrix}.$$

After an odd-even permutation of rows and columns of (9), we obtain a Lanczos factorization that contains both LBD factorizations (4) and (5):

$$\begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}\begin{pmatrix} U_{k+1} & 0 \\ 0 & V_k \end{pmatrix} = \begin{pmatrix} U_{k+1} & 0 \\ 0 & V_k \end{pmatrix}\begin{pmatrix} 0 & B_k \\ B_k^* & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \alpha_{k+1}v_{k+1}e_{k+1}^* & 0 \end{pmatrix}. \tag{10}$$

We next discuss some of the difficulties of the LBD algorithm. An important difficulty with LBD, typical of Lanczos type algorithms, is the loss of orthogonality among the basis vectors in $V_k$ and $U_{k+1}$ [36]. The application of reorthogonalization schemes can remedy the problem, though this is at an extra computational cost. A compromise is to use partial reorthogonalization schemes that dynamically update the level of orthogonality among the basis vectors at each step. Recent work of Larsen has produced MATLAB codes that implement partial reorthogonalization in the context of LBD; see [28–31] as well as [43,44]. When the matrix at hand is very large, in order to obtain acceptable approximations to the smallest singular triplets, even using sophisticated schemes for partial reorthogonalization, convergence can be slow and the bases $U_{k+1}$, $V_k$ need to become so large that computational and storage costs become overwhelming.

As we show in the next sections, to address these problems, we incorporate implicit restarting mechanisms in LBD that maintain computational and memory requirements constant at each step. Furthermore, we combine implicit restarting with harmonic Ritz values for the approximation of the smallest singular triplets.

## 3. Implicitly restarted LBD

Implicit restarting, proposed by Sorensen in [47] for the Arnoldi and Lanczos iterations, through its practical implementation in ARPACK [32], is widely acknowledged to be one of the most successful frameworks for solving very large eigenproblems. In this section we describe how to apply implicit restarting in the context of LBD. Implicit restarting in LBD was first studied in [5] and was later combined with partial reorthogonalization in [28,31].

In Section 2 we established that LBD is equivalent to Lanczos applied on $AA^*$, according to factorization (7). Therefore, after $l = k + p$ steps of LBD we can apply $p$ implicitly shifted QR steps on matrix $T_l = B_l B_l^*$, which is real symmetric and tridiagonal. Alternatively, we can apply Golub–Kahan SVD steps [17, Section 8.6.2] directly on the bidiagonal matrix $B_l$ in order to enhance stability [39]. The implicitly shifted QR step is applied directly on an upper bidiagonal matrix by means of bulgechasing as shown in Algorithm 2. The first Givens rotation (line 4) creates a "bulge" (i.e., a nonzero element in the subdiagonal) and the trailing Givens rotations "chase" the bulge out of the matrix in order to restore its upper bidiagonal form. Since we work with a lower bidiagonal matrix, the update can be written as $B_l^+ = Q_L B_l Q_R^*$, where $Q_L \in \mathbb{R}^{(l+1)\times(l+1)}$ and $Q_R \in \mathbb{R}^{l\times l}$ are orthogonal matrices that implement Givens rotations. Therefore, by updating the bases $V_l$ and $U_{l+1}$ we can recover the bidiagonalization

$$AV_k^+ = U_{k+1}^+ B_k^+,$$

where $V_k^+ = V_l Q_R(1:k,:)^*$ and $U_{k+1}^+ = U_{l+1} Q_L(1:k+1,:)^*$. This updated LBD factorization is what we would have obtained after $k$ steps of LBD with the special starting vector

$$u_1^+ = \left(AA^* - \mu^2 I\right)u_1,$$

using shift $\mu$.

**Algorithm 2** (*Bulgechasing*) *Golub–Kahan SVD step* [17, Section 8.6.2].

*Input*: Tridiagonal matrix $T_l = B_l B_l^*$, implicit shift $\mu$
*Output*: Updated upper bidiagonal matrix $B_l^+$

1. Set $y = t_{1,1} - \mu$ and $z = t_{1,2}$
2. **for** $i = 1 : l-1$
3. Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that
   $(y, z)\left(\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right) = (*, 0)$
4. Apply to $B_l$ the Givens rotation from the right: $B_l = B_l G(i, i+1, \theta)$
5. Set: $y = b_{i,i}$ and $z = b_{i+1,i}$
6. Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that
   $\left(\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right)^\top \binom{y}{z} = \binom{*}{0}$
7. Apply to $B_l$ the Givens rotation from the left $B_l = G(i, i+1, \theta)^\top B_l$
8. **if** $i < l-1$ **then**
9. Set: $y = b_{i,i+1}$ and $z = b_{i,i+2}$
10. **end**
11. **end**

If the previous procedure is repeated for $p-1$ shifts, say $\mu_2, \mu_3, \ldots, \mu_p$, we obtain a bidiagonalization that corresponds to the starting vector

$$u_1^+ = \prod_{i=1}^{p} \big(AA^* - \mu_i^2 I\big) u_1,$$

and therefore we can apply polynomial filtering with implicit restarts of LBD as an equivalent to implicitly restarted Lanczos on $AA^*$.

We also showed in relation (10) that LBD is equivalent to Lanczos applied on the augmented matrix $C$. It is thus natural to ask whether implicitly restarted LBD can be equivalent to implicitly restarted Lanczos on $C$. If the implicit restarting mechanism is applied directly on $C$, the updated starting vector $q_1^+ = \prod_{i=1}^{p}(C - \mu_i I)q_1$ will not have, in general, the special structure $q_1^+ = [u_1^+; 0]$ and therefore it will not be possible to extract an LBD decomposition from it. This is shown in the following proposition that is stated assuming exact arithmetic.

**Proposition 1.** *It is not possible, in general, to apply implicit QR steps on the Lanczos factorization* (9) *of the augmented matrix $C$, and obtain a Lanczos factorization that can be computed by LBD.*

**Proof.** Implicit restarts essentially perform polynomial filtering on the starting vector $u_1$. After $p$ implicit QR steps on factorization (9), the updated Lanczos factorization can be written as

$$C Q_{2k}^+ = Q_{2k}^+ T_{2k}^+ + \beta_{k+1}^+ q_{2k+1}^+ e_{2k}^*,$$

with starting vector $q_1^+ = \pi(C)q_1$ where $\pi(C)$ is a non-trivial polynomial of the augmented matrix $C$ of degree $p$. Observe now that the powers of $C$ have the following special structure

$$C^{2i} = \begin{bmatrix} (AA^*)^i & 0 \\ 0 & (A^*A)^i \end{bmatrix}, \qquad C^{2i+1} = \begin{bmatrix} 0 & A(A^*A)^i \\ A^*(AA^*)^i & 0 \end{bmatrix}, \quad i = 1, 2, \ldots.$$

If we define the polynomials $\pi_o$ and $\pi_e$ containing strictly odd and even powers respectively such that $\pi(C) = \pi_o(C) + \pi_e(C)$, then for the polynomial $\pi(C)$ it holds that

$$\pi(C) = \begin{bmatrix} \pi_o(AA^*) & A\pi_e(A^*A) \\ A^*\pi_e(AA^*) & \pi_o(A^*A) \end{bmatrix}.$$

Since for the starting vector it holds that $q_1^* = [u_1^*, 0]$, we have that

$$q_1^+ = \begin{bmatrix} \pi_o(AA^*)u_1 \\ A^*\pi_e(AA^*)u_1 \end{bmatrix}.$$

Observe now that according to (2) it holds that

$$\pi_e(AA^*) = U\pi_e(\Lambda)U^*, \quad \Lambda = \operatorname{diag}\big(\sigma_1^2, \ldots, \sigma_m^2\big),$$

and thus $\|A^*\pi_e(AA^*)u_1\|_2 = \|A^*U\pi_e(\Lambda)U^*u_1\|_2 = \|V\Sigma^*\pi_e(\Lambda)U^*u_1\|_2$, where we have used the SVD of $A$. Since $V$ is orthonormal, if we denote by $\Sigma_1 = \operatorname{diag}[\sigma_1, \ldots, \sigma_m]$, it follows that

$$\big\|A^*\pi_e\big(AA^*\big)u_1\big\|_2 = \big\|\Sigma^*\pi_e(\Lambda)U^*u_1\big\|_2 = \big\|[\Sigma_1\ 0]^*\pi_e\big(\Sigma_1^2\big)U^*u_1\|_2.$$

Notice that $U^*u_1$ cannot be zero since $U$ is orthonormal and has full rank. Furthermore, for a general matrix with $m$ distinct nonzero singular values, the above norm would be zero only if $\pi_e(\sigma_i^2) = 0$ for

$i = 1, \ldots, m$. Since the degree of $\pi_e$ is $p < m$, however, this can only happen if $\pi_e$ is identically zero. Therefore, in general, the updated vector $q_1^+$ cannot have the special structure $q_1^+ = [u_1^+; 0]$, thus the updated Lanczos factorization on the augmented matrix $C$ cannot be equivalent to an LBD factorization. □

## 4. Shift selection strategies

We next consider shift selection for the implicitly restarted LBD. In particular, we examine two strategies: (i) exact Ritz values and (ii) exact harmonic Ritz values. Alternative shift strategies include the zeros of Chebyshev polynomials and Leja points (see, e.g., [10] and references therein).

### *4.1. Ritz values*

Using relation (5) and premultiplying with $U_l^*$ we see that after $l = k + p$ steps of LBD, the following relationship holds:

$$U_l^* A A^* U_{l+1} = U_l^* A V_l B_l + \alpha_{l+1} U_l^* A v_{l+1} e_{l+1}^*.$$

Applying relation (4) and considering only the first $l$ columns of each side it follows that $U_l^* A A^* U_l = \widehat{B}_l \widehat{B}_l^*$, where $\widehat{B}_l \in \mathbb{R}^{l \times l}$ denotes the square lower subdiagonal matrix that we obtain by omitting the last row of $B_l$. Therefore, the squares of the singular values of the matrix $\widehat{B}_l$ are Ritz values of the Hermitian matrix $AA^*$ and therefore provide approximations to the singular values of $A$. Our exact Ritz values strategy is to pick as implicit shifts the largest $p$ of the squared singular values of $\widehat{B}_l$. It is worth noting that since our target is to compute singular values of $\widehat{B}_l$ and not eigenvalues of $\widehat{B}_l \widehat{B}_l^*$, we do not expect loss of precision due to squared conditioning. Furthermore, by not approximating squared singular values we do not aggravate any existing clustering of the smallest singular values of $A$.

### *4.2. LBD and harmonic Ritz values*

Ritz values readily provide a straightforward shift strategy. It is often the case, however, that the smallest singular values of $A$ are clustered. This is a situation that can significantly slow down the convergence of implicitly restarted Lanczos. In order to secure satisfactory convergence rates we can try to approximate the smallest singular values of $A$ by computing the largest Ritz values of $(AA^*)^{-1}$. In the remainder of this section we will be assuming that $A$ has full rank. In line with the matrix-free approach aspired to in this paper, however, we wish to avoid explicit computations with $(AA^*)^{-1}$. This becomes possible using the concept of harmonic Ritz values [38]:

**Definition 2.** A value $\tilde{\theta}_k \in \mathbb{C}$ is a harmonic Ritz value of a matrix $G \in \mathbb{C}^{m \times m}$ with respect to some linear subspace $\mathcal{W}_k$ if $\tilde{\theta}_k^{-1}$ is a Ritz value of $G^{-1}$ with respect to $\mathcal{W}_k$.

Returning to the Lanczos factorization (7), since we are interested in the Ritz values of $(AA^*)^{-1}$ we could compute harmonic Ritz values of $AA^*$. We do this by means of oblique projection and the corresponding Petrov–Galerkin condition. Our presentation in the remainder of this section owes a lot to the discussion of Sleijpen and van der Vorst in [45] regarding harmonic Ritz values (the reader can also

refer to [21] for a relevant discussion). In particular, if the search space $\mathcal{U}_{l+1}$ is of dimension $l+1$ and the test space is $\mathcal{W}_{l+1} = AA^*\mathcal{U}_{l+1}$ then the corresponding Petrov–Galerkin condition becomes

$$AA^*\tilde{u}_{l+1} - \tilde{\theta}_{l+1}\tilde{u}_{l+1} \perp AA^*\mathcal{U}_{l+1},$$

where $\tilde{\theta}_{l+1}$ is a harmonic Ritz value of $AA^*$. Furthermore, if $U_{l+1}$ and $W_{l+1}$ are bases that span the subspaces $\mathcal{U}_{l+1}$ and $\mathcal{W}_{l+1}$, respectively, then the harmonic Ritz values of $AA^*$ are the eigenvalues of matrix $\widetilde{H}_{l+1}$:

$$\widetilde{H}_{l+1} = \left(W_{l+1}^* U_{l+1}\right)^{-1} W_{l+1}^* AA^* U_{l+1}. \tag{11}$$

It should be clear now how to compute the shifts for the implicit restart. At each restart we compute the harmonic Ritz values and use as shifts the $p$ largest ones. It is worth noting that we are actually using an "exact shift" strategy with harmonic rather than ordinary Ritz values. As we show next, the harmonic Ritz values can be easily obtained at the cost of an additional step of the LBD and the SVD of the corresponding lower bidiagonal matrix.

**Proposition 3.** *The harmonic Ritz values $\tilde{\theta}$ and vectors $\tilde{y}$ of matrix $AA^*$ with respect to the subspace $\mathcal{W}_{l+1}$ correspond to the eigenvalue problem*

$$B_{l+1}^* B_{l+1} y = \tilde{\theta} y, \quad y = \widehat{B}_{l+1}^* \tilde{y}, \tag{12}$$

*where $B_{l+1}$ is the $(l+2)\times(l+1)$ lower bidiagonal matrix of the LBD of length $l+1$.*

**Proof.** Using relations (4), (5) we have

$$AA^* U_{l+1} = U_{l+2} B_{l+1} \widehat{B}_{l+1}^*,$$

where $B_{l+1}$ is the $(l+2)\times(l+1)$ lower bidiagonal matrix that corresponds to the LBD of length $l+1$ and $\widehat{B}_{l+1}$ is derived by deleting the last row of $B_{l+1}$. If we define the matrix $T_{l+1} = B_{l+1}\widehat{B}_{l+1}^*$ then, according to [45], the harmonic Ritz values are eigenvalues of the generalized eigenvalue problem

$$T_{l+1}^* T_{l+1} \tilde{y} = \tilde{\theta} \widehat{T}_{l+1}^* \tilde{y}, \tag{13}$$

where $\widehat{T}_{l+1}$ is obtained by deleting the last row of $T_{l+1}$. However, substituting $T_{l+1}$ and $\widehat{T}_{l+1}$ in (13) we have

$$\widehat{B}_{l+1} B_{l+1}^* B_{l+1} \widehat{B}_{l+1}^* \tilde{y} = \tilde{\theta} \widehat{B}_{l+1} \widehat{B}_{l+1}^* \tilde{y}.$$

Assuming that $\widehat{B}_{l+1}$ is nonsingular and setting $y = \widehat{B}_{l+1}^* \tilde{y}$ we have that

$$B_{l+1}^* B_{l+1} y = \tilde{\theta} y. \quad \square$$

Therefore, the harmonic Ritz values sought in this section are equal to the eigenvalues of the matrix $B_{l+1}^* B_{l+1}$ and can be computed directly from the singular values of the $(l+2)$ by $(l+1)$ lower bidiagonal matrix $B_{l+1}$. Note that an additional step of LBD is required in order to compute $B_{l+1}$.

Observe that in "pure" LBD nothing new is achieved by invoking the harmonic Ritz values instead of the standard Ritz values technique. However, this is not the case for implicitly restarted LBD. As shown by Morgan in [35, Theorem 5.14] the subspace generated by implicitly restarted Arnoldi on a matrix $G$ using the unwanted harmonic Ritz values as shifts is

$$\operatorname{span}\{r, Gr, G^2r, G^3r, \ldots, G^{l-k-1}r, \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_k\},$$

where the $\tilde{y}_j$'s are the harmonic Ritz vectors corresponding to the desired harmonic Ritz values, whereas when using the unwanted Ritz values as shifts, the $\tilde{y}_j$'s are the Ritz vectors corresponding to the desired Ritz values. Note that the two subspaces are distinct because of the different set of approximate eigenvectors in the added portion; hence the convergence behavior of implicitly restarted LBD will not necessarily be the same for the two cases. In Section 8.1 we provide numerical evidence illustrating the superior behavior of harmonic Ritz values vis-à-vis ordinary Ritz values.

## 5. Deflation

One important issue in the design of implicitly restarted Arnoldi algorithms is the implementation of efficient deflation techniques that enhance convergence and stability and provide an effective way to compute multiple and clustered eigenvalues. This is so as to let the methods become an effective alternative to block methods. It is worth noting that implicitly restarted Arnoldi has been combined with block methods to deal with the computation of few selected eigenpairs and singular triplets in an algorithm recently proposed by Baglama et al. [1]. We thus need to consider how to implement deflation in the context of implicitly restarted LBD. Our scheme builds upon results presented in [2,33,46]. As in [33] we employ "locking", that decouples converged approximate singular values and singular subspaces. In this section we describe the modification and application of the "orthogonal deflating transformation" (ODT for short), a scheme originally proposed by Sorensen in [46] in the context of implicitly restarted Arnoldi for eigenvalues. We show that the transformation can be applied directly on the bidiagonal matrix that results from implicitly restarted LBD. The deflation scheme enables the stable and efficient locking of approximate singular values that have converged with relative accuracies that may be much inferior to the machine precision.

The ODT is based upon a special unitary matrix, say $Q$, that is built, as shown in [46] to satisfy $Qe_1 = y$ for a suitably chosen unit norm vector $y = [\eta_1, \ldots, \eta_n]^*$; cf. [2,46] for the construction of $Q$. Furthermore, $Q$ has the form

$$Q = R + ye_1^*, \quad \text{with } Re_1 = 0,$$

where $R$ is upper triangular, its first column is zero and $R^*y = 0$. It may also be written as

$$Q = L + yg^*, \quad \text{with } Le_1 = 0,\ L^*y = e_1 - g,$$

where $L$ is lower triangular and $g^* = e_1^* + \frac{1}{\eta_1}e_1^*R$. Assuming now that such a $Q$ can be built, the following lemma shows how to apply the ODT in the case of implicitly restarted LBD.

**Lemma 4.** *Let $(\theta, y_L, y_R)$ be an approximate singular triplet of $A \in \mathbb{C}^{m\times n}$ computed from the lower bidiagonal matrix $B$ resulting after $k$ steps of LBD. Let also $Q_L = Q_L(y_L) \in \mathbb{C}^{(k+1)\times(k+1)}$ and $Q_R = Q_R(y_R) \in \mathbb{C}^{k\times k}$ be the unitary matrices produced for ODT from the vectors $y_L$ and $y_R$, respectively. Then the updated matrix $\breve{B} = Q_L^* B Q_R$ is lower bidiagonal and has the special form $\breve{B} = \left(\begin{smallmatrix}\theta & 0\\ 0 & \widehat{B}\end{smallmatrix}\right)$ where $\theta$ is the approximate singular value and $\widehat{B}$ is also lower bidiagonal.*

**Proof.** Using the same notation as above, the following relations hold for $Q_L$:

$$Q_L e_1 = y_L, \tag{14}$$
$$Q_L = R_L + y_L e_1^*, \qquad R_L e_1 = 0, \qquad R_L^* y_L = 0, \tag{15}$$
$$Q_L = L_L + y_L g_L^*, \qquad L_L e_1 = 0, \qquad L_L^* y_L = e_1 - g_L. \tag{16}$$

Similarly, the following relations also hold for $Q_R$:

$$Q_R e_1 = y_R, \tag{17}$$
$$Q_R = R_R + y_R e_1^*, \qquad R_R e_1 = 0, \qquad R_R^* y_R = 0, \tag{18}$$
$$Q_R = L_R + y_R g_R^*, \qquad L_R e_1 = 0, \qquad L_R^* y_R = e_1 - g_R. \tag{19}$$

We will prove that $\breve{B} = Q_L^* B Q_R$ is upper Hessenberg as well as lower triangular, and therefore lower bidiagonal. In particular,

$$\begin{aligned}\breve{B} &= Q_L^* B Q_R = Q_L^* B\big(R_R + y_R e_1^*\big) = Q_L^* B R_R + Q_L^*(B y_R) e_1^* \\ &= Q_L^* B R_R + Q_L^* \theta y_L e_1^* = Q_L^* B R_R + \theta e_1 e_1^*,\end{aligned}$$

since $Q_L^* y_L = e_1$. Therefore,

$$\begin{aligned}\breve{B} &= Q_L^* B Q_R = \big(L_L^* + g_L y_L^*\big) B R_R + \theta e_1 e_1^* = L_L^* B R_R + g_L\big(y_L^* B\big) R_R + \theta e_1 e_1^* \\ &= L_L^* B R_R + \theta g_L\big(y_R^* R_R\big) + \theta e_1 e_1^* = L_L^* B R_R + \theta e_1 e_1^*,\end{aligned}$$

since $y_R^* R_R = 0$; cf. relations (18). Matrix $L_L^* B R_R + \theta e_1 e_1^*$ is upper Hessenberg because $L_L^*$ and $R_R$ are upper triangular and $B$ is lower bidiagonal, thus $\breve{B}$ is upper Hessenberg. Furthermore,

$$\begin{aligned}\breve{B} &= Q_L^* B Q_R = Q_L^* B\big(L_R + y_R g_R^*\big) = Q_L^* B L_R + Q_L^*(B y_R) g_R^* \\ &= Q_L^* B L_R + \theta Q_L^* y_L g_R^* = Q_L^* B L_R + \theta e_1 g_R^*,\end{aligned}$$

since $Q_L^* y_L = e_1$ because of (14). Therefore,

$$\begin{aligned}\breve{B} &= \big(R_L + y_L e_1^*\big)^* B L_R + \theta e_1 g_R^* = \big(R_L^* + e_1 y_L^*\big) B L_R + \theta e_1 g_R^* \\ &= R_L^* B L_R + e_1\big(y_L^* B\big) L_R + \theta e_1 g_R^* = R_L^* B L_R + \theta e_1 y_R^* L_R + \theta e_1 g_R^* \\ &= R_L^* B L_R + \theta e_1\big(y_R^* L_R + g_R^*\big) = R_L^* B L_R + \theta e_1 e_1^*,\end{aligned}$$

since $y_R^* L_R + g_R^* = e_1^*$ because of (19). Since both $R_L^*$ and $L_R$ are lower triangular, $B$ would be lower bidiagonal while the rank-one update would not modify the lower triangular form, therefore $\breve{B}$ is also lower triangular. □

It is worth noting that the observations concerning the numerical stability of ODT discussed in [46] carry over to the present case. In particular, note that matrices $Q_L, Q_R$ are built from $y_L$ and $y_R$, respectively, therefore, some of their implicit properties are not exactly satisfied in finite precision arithmetic. Therefore, in order for $\breve{B} = Q_L^* B Q_R$ to be numerically upper Hessenberg, special care must be taken so that $\|g_L(y_L^* B) R_R\|_2$ would remain small in practice. If we write $y_L^* B = \theta y_R^* + z^*$, where $z$ denotes numerical error, then it follows that $\|g_L(y_L^* B) R_R\|_2 = \frac{1}{\eta_1^L}\|z^* R_R\|_2$, where $\eta_1^L$ denotes the first component of $y_L$. Unfortunately, for small values of $\eta_1^L$ the above factor could be large and a rescaling strategy, such as the one described in [46], must be applied. On the other hand, $R_L^* B L_R + \theta e_1(y_L^* B L_R + \theta g_R^*) = R_L^* B L_R + \theta e_1 e_1^* + e_1 z^* L_R$. Since $L_R = Q_R - y_R g_R^*$, if we apply the aforementioned rescaling strategy, the norm $\|g_R\|_2 = \frac{1}{\eta_1^R}$ is kept small and therefore $\breve{B}$ would be numerically lower triangular since $\|e_1 z^* L_R\|_2$ will be small for small $z$.

## 6. Refined singular vector approximations

It is often the case when computing eigenvalues that a Ritz vector may exhibit poor convergence even though the corresponding Ritz value has converged. Jia proposed in [24] a refined Ritz vector strategy. The key is to approximate the eigenvector by means of a refined Ritz vector designed to minimize the norm of the residual over the subspace involved. In Section 2 we saw that the LBD decompositions are equivalent to Lanczos decompositions on either $AA^*$ or the augmented matrix $C$ (with a starting vector of special structure). Therefore, we can compute the refined residual (and vector) using either $C$ or $AA^*$.

We first outline the refinement process for matrix $C$. Given $\tilde{\sigma}$ the approximation to the smallest singular value of $A$, we seek the refined singular vectors $\tilde{u} = U_{l+1}s \in \mathcal{K}_{l+1}(AA^*, u_1)$ and $\tilde{v} = V_l t \in \mathcal{K}_l(A^*A, v_1)$ that solve the joint minimization problem:[2]

$$
\begin{aligned}
&\min_{\substack{s\in\mathbb{C}^{l+1},t\in\mathbb{C}^l\\ \|[s;t]\|_2=1}} \left\| \left[ \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix} - \tilde{\sigma} I_{m+n} \right] \begin{pmatrix} U_{l+1}s \\ V_l t \end{pmatrix} \right\|_2 \\
&\quad = \min_{\substack{s\in\mathbb{C}^{l+1},t\in\mathbb{C}^l\\ \|[s;t]\|_2=1}} \left\| \left[ \begin{pmatrix} 0 & AV_l \\ A^*U_{l+1} & 0 \end{pmatrix} - \tilde{\sigma} \begin{pmatrix} U_{l+1} & 0 \\ 0 & V_l \end{pmatrix} \right] \begin{pmatrix} s \\ t \end{pmatrix} \right\|_2 \\
&\quad = \min_{\substack{s\in\mathbb{C}^{l+1},t\in\mathbb{C}^l\\ \|[s;t]\|_2=1}} \left\| \begin{pmatrix} U_{l+1} & 0 \\ 0 & V_{l+1} \end{pmatrix} \left[ \begin{pmatrix} 0 & B_l \\ B_l^* & 0 \\ \alpha_{l+1}e_{l+1}^* & 0 \end{pmatrix} - \tilde{\sigma} \begin{pmatrix} I_{2l+1} \\ 0 \end{pmatrix} \right] \begin{pmatrix} s \\ t \end{pmatrix} \right\|_2 \\
&\quad = \min_{\substack{s\in\mathbb{C}^{l+1},t\in\mathbb{C}^l\\ \|[s;t]\|_2=1}} \left\| \left[ \begin{pmatrix} 0 & B_l \\ B_l^* & 0 \\ \alpha_{l+1}e_{l+1}^* & 0 \end{pmatrix} - \tilde{\sigma} \begin{pmatrix} I_{2l+1} \\ 0 \end{pmatrix} \right] \begin{pmatrix} s \\ t \end{pmatrix} \right\|_2 \\
&\quad = \sigma_{\min}(\mathbf{R}_{2l+2}),
\end{aligned}
$$

where

$$
\mathbf{R}_{2l+2} = \begin{pmatrix} 0 & B_l \\ B_l^* & 0 \\ \alpha_{l+1}e_{l+1}^* & 0 \end{pmatrix} - \tilde{\sigma} \begin{pmatrix} I_{2l+1} \\ 0 \end{pmatrix}, \tag{20}
$$

since the norm of the residual is minimized when $[s^*\ t^*]^*$ is the right singular vector associated with the smallest singular value $\sigma_{\min}(\mathbf{R}_{2l+2})$. This singular value is called the refined residual. In general, it is known that the angle between the refined Ritz vector and the exact eigenvector is better than the corresponding angle between the latter and the standard Ritz vector. Furthermore, notice that we can use the Rayleigh quotient $\rho = \tilde{u}^* A \tilde{v}$ in an attempt to obtain an improved eigenvalue, since $\rho$ may be more accurate than $\tilde{\sigma}$; cf. [48, Section 4.3].

Concerning matrix $AA^*$, decomposition (7) suggests that if $\tilde{\sigma}_{\min}$ is the current approximation to the smallest singular value of $A$, the refined residual and refined singular vector can be retrieved as before, that is by computing the smallest singular value and right singular vector of

$$
\mathbf{B}_{l+1} = \begin{pmatrix} B_l B_l^* + \alpha_{l+1}^2 e_{l+1}e_{l+1}^* \\ \beta_{l+2}\alpha_{l+1}e_{l+1}^* \end{pmatrix} - \tilde{\sigma}_{\min}^2 \tilde{I}. \tag{21}
$$

[2] We thank a referee for suggesting this presentation of the refined residual.

We next have to decide which refined residual to compute, the one from $AA^*$ or from $C$? Since (21) involves the tridiagonal matrix $BB^*$ one might expect stability problems in contrast to (20). Furthermore, the refined residual for $AA^*$ yields approximations only to the left singular vector so that to obtain approximations to the right singular vector we would need to use the relation $v_{\min} = \frac{1}{\sigma}_{\min} A^* u_{\min}$ or also work with the refined residual of $A^*A$. It is thus preferable to use the augmented matrix $C$ which also facilitates the concurrent approximation of both the left and right singular vectors of $A$. For more details, see also the discussion in [48, Section 4.3].

## 7. IRLANB: Implicitly restarted harmonic Lanczos bidiagonalization

Based on the previous discussion, we next display our proposed method as Algorithm 3. Parameter $l = k + p$ is the maximum dimension of the bidiagonalization, where $p$ is the number of implicitly shifted QR steps applied on $B_l$. Parameter `eignum` determines the number of smallest singular values that we seek and `tol` controls the convergence tolerance. The first step of `IRLANB` constructs an LBD factorization of length $l$. For this purpose we have used the function `lanbpro` from Larsen's `PROPACK` [28] (see also [29]) which is a set of `MATLAB` codes for the Hermitian eigenvalue and SVD problems based on Lanczos and Lanczos bidiagonalization with partial reorthogonalization. It has been observed experimentally that implicit restarting still works when the Lanczos vectors are semi-orthogonal [31, pp. 19–20]. As described in Section 4, if we select to shift with Ritz values, we prefer, for reasons of stability, to compute singular values of $B_l$ rather than eigenvalues of $B_l B_l^*$. If, instead, we select to shift by means of harmonic Ritz values, we could use the singular values of $B_{l+1}$.

**Algorithm 3** (`IRLANB`). An implicitly restarted Lanczos bidiagonalization method to compute a few of the smallest singular triplets of large sparse matrices.

*Input*: matrix $A \in \mathbb{C}^{m\times n}$, $k$, $p$, `eignum`, `tol`. Starting vector $u_1$. Set $l = k + p$
*Output*: `eignum` of the smallest singular triplets

1. Compute bases $U_{l+1}$ and $V_l$ and bidiagonal $B_l$ using `LBD`
2. **Repeat**
3. **if** (*shifts == Ritz*) **then**
4. Compute the singular values $\sigma_i$, $i = 1, \ldots, l$ of $\widehat{B}_l$
5. **elseif** (*shifts == Harmonic*)
6. Compute $B_{l+1}$ by an additional step of LBD and the singular values $\sigma_i$, $i = 1, \ldots, l+1$ of $B_{l+1}$
7. **end**
8. Perform $p$ implicit QR steps using bulgechasing on $B_l$ with the $p$ largest $\sigma_i^2$ as shifts and update the LBD factorization: $AV_k^+ = U_{k+1}^+ B_k^+$
9. Compute the approximation $\tilde{\sigma}_{\min}(A) = \min\{\sigma_i\}$
10. Compute the refined residual $r$ of $\tilde{\sigma}_{\min}(A)$
11. **if** $\|r\| \leqslant$ `tol` $*$ `normest`$(A)$ **then**
12. Compute the left and right refined singular vectors of $\tilde{\sigma}_{\min}$
13. Compute $Q_L$ and $Q_R$ matrices using ODT and perform deflation

14. Discard the first column of $U_{l+1}$, $V_l$ and the first row and column of $B_l$
15. $k = k - 1$ and `eignum` $=$ `eignum` $-1$
16. **end**
17. Reorthogonalize $u_{k+1}^+$ and $v_k^+$ against all previous (even converged) basis vectors
18. Extend $AV_k^+ = U_{k+1}^+ B_k^+$ to length $l = k + p$ using `LBD`
19. **Until** *convergence* of all `eignum` singular values

The next step is to compute the 2-norm of the refined residual according to either one of the strategies described in Section 6. If this norm is smaller than `tol` scaled by an estimation (`normest(`$A$`)`) of $\|A\|$, then the approximation to the singular value has converged. In practice, we use as `normest(`$A$`)` the largest singular value of $B_l$ before the first restart of `IRLANB`. On the other hand, if convergence has not taken place we proceed to the reorthogonalization steps (line 17) and repeat the process. As soon as the current approximation to $\sigma_{\min}$ satisfies the convergence criterion we compute the corresponding left and right refined singular vectors and proceed with the deflation procedure. We compute the orthogonal matrices $Q_L$ and $Q_R$ using ODT, as described in Section 5. Purging is accomplished by discarding the first column of the bases $U_{k+1}^+$ and $V_k^+$ as well as the first row and column of $B_k^+$. As a result, we obtain an LBD factorization of length $(k-1)$ while the deflated factorization no longer contains the targeted singular values. However, in subsequent restarts we reorthogonalize the updated vectors $u_{k+1}^+$ and $v_k^+$ against all previous vectors, even purged ones, since roundoff may introduce components towards the directions of converged vectors. Note that since we are computing a small number of singular triplets, the extra cost incurred is low. Computational practice indicates that this limited reorthogonalization suffices to maintain an acceptable level of orthogonality among basis vectors that may have been degraded by the implicit restart.

## 8. Numerical experiments

In this section we present numerical experiments designed to illustrate the numerical and computational performance of `IRLANB`. All codes were written in `MATLAB 6.1` and ran on a 866 MHz Pentium III equipped with 1 GB of RAM and 512 Kb of cache memory running `Windows 2000 Server`. We also illustrate the performance of `IRLANB` vs. two recent methods for which `MATLAB` codes are publicly available and which are matrix-free, so as to permit the solution of very large sparse problems in computational environments such as the above. These methods were:

*`IRBLSVDS-IRBLEIGS:`* Code due to Baglama, Calvetti, and Reichel and based on implicitly restarted block Lanczos [1] designed to compute one or more eigenvalues and/or singular values.[3]

*`JDQZ:`* Code based on Jacobi–Davidson QZ method due to Fokkema, Sleijpen and van der Vorst and implemented in `MATLAB` [14].[4]

Note that if asked to compute a few of the smallest singular values of sparse matrices, the `MATLAB 6` built-in function `svds`, that is based on a compiled implementation of `ARPACK` (`eigs`), applies shift-

[3] At http://hypatia.math.uri.edu/~jbaglama.

[4] At http://www.math.ruu.nl/people/sleijpen/JD_software/JDQZ.html.

and-invert and requires an LU decomposition of the augmented matrix $C$. Therefore, we do not include `svds` in our experiments. It is also worth noting that in [1], `IRBLSVDS-IRBLEIGS` was compared to methods selected based on criteria similar to the ones described herein. We note that in all the experiments that follow we employed the pure `MATLAB` version of `PROPACK` in which no `mex` files are used.

### 8.1. Ritz and harmonic Ritz shift strategies

The first set of experiments is designed to illustrate the convergence behavior of Ritz values vis-a-vis that of harmonic Ritz values, when used as shifts in the implicitly restarted LBD algorithm. We constructed a sequence of diagonal matrices $\mathtt{A_s} \in \mathbb{R}^{n\times n}$, $n = 100$, $\mathtt{s} = 1, 2, \ldots$, that exhibit increasing clustering of their smallest singular values. In `MATLAB` notation:

$$\mathtt{A_s} = \mathtt{spdiags}([\mathtt{1:10\hat{}(-s)}:\mathtt{1}+\mathtt{9}*\mathtt{10\hat{}(-s)}, \mathtt{2:1:100}]', \mathtt{0}, \mathtt{100}, \mathtt{100}). \tag{22}$$

The test space dimension was $l = k + p = 20$ while at each restart we performed $p = 10$ implicit QR steps. We used a random starting vector normalized to have unit length and convergence tolerance $\mathtt{tol} = 1\mathrm{e}{-8}$. Fig. 1 illustrates the true relative errors for Ritz as well as for harmonic Ritz shifts. It is evident that as the clustering of the smallest singular values of $\mathtt{A_s}$ increases, harmonic Ritz values either converge significantly faster (they require fewer restarts) or, with the same backward error (2-norm of residual) used for the Ritz values, produce results with better forward relative error. Therefore, in case of severe clustering of the smallest singular values we use harmonic Ritz values, which, as shown in Section 4.2, can be computed at relatively small cost.

### 8.2. Experiments with ill-conditioned matrices

We next investigate the behavior of `IRLANB` with harmonic Ritz values and ill-conditioned matrices. We constructed a sequence of dense matrices $\mathtt{A_s} \in \mathbb{R}^{n\times n}$, $n = 100$, $\mathtt{s} = 4, \ldots, 7$ and $9, \ldots, 12$ with increasing condition numbers:

$$\mathtt{H_s} = \mathtt{spdiags}(\mathtt{linspace}(\mathtt{1}, \mathtt{10\hat{}s}, \mathtt{100})', \mathtt{0}, \mathtt{100}, \mathtt{100}),$$
$$\mathtt{A_s} = \mathtt{orth}(\mathtt{rand}(\mathtt{100})) * \mathtt{H_s} * \mathtt{orth}(\mathtt{rand}(\mathtt{100}))'.$$

We used the same starting vector and parameters as in the previous examples ($k = 20$, $p = 10$), except for the convergence tolerance which was set to $\mathtt{tol} = 1\mathrm{e}{-12}$. Fig. 2 illustrates the absolute value of the relative error achieved by `IRLANB`. For the cases $s = 4, 5, 6, 7$, `IRLANB` computed the smallest singular value with relative error smaller than $10^{-10}$. For even higher values of the condition number ($s = 9, 10, 11, 12$), convergence clearly deteriorates. In particular, for the most difficult case of $\kappa(A) = 10^{-12}$, the relative error is approximately O(1), in agreement with the predicted forward error bound. On the other hand, when we used a smaller convergence tolerance, specifically $\mathtt{tol} = 10^{-14}$, `IRLANB` converged with three correct decimal digits (absolute relative error approximately $0.5 \times 10^{-4}$).

We next include a numerical example illustrating the behavior of the refined residual as opposed to the Ritz one. We used matrix `illc1850` of dimension $1850 \times 712$ and $\kappa(A) = 1.4 \times 10^3$ from Matrix Market,[5] with parameters $k + p = 50$, $p = 30$ and $\mathtt{tol} = 1\mathrm{e}{-8}$. The right diagram in Fig. 3 shows the

[5] At http://math.nist.gov/MatrixMarket.

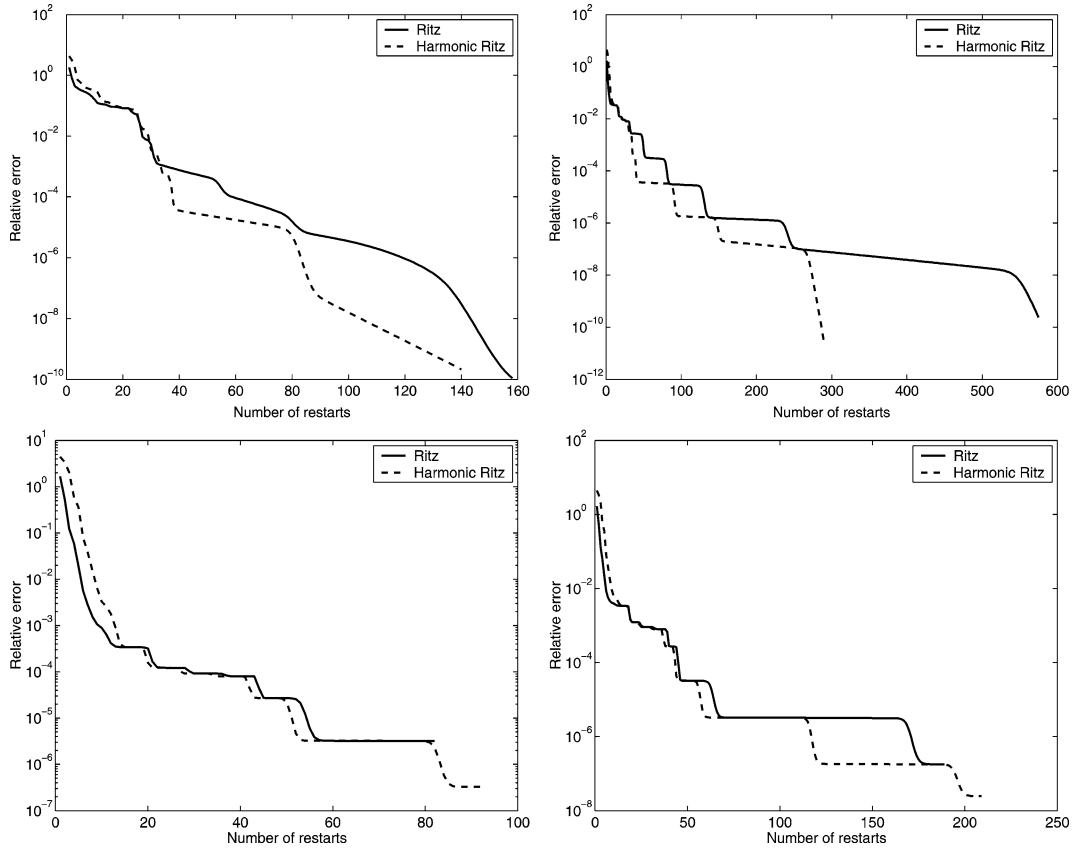


Fig. 1. Experiments with the diagonal matrices (22). Starting from the left top corner and moving clockwise we depict relative errors for $s = 1, 2, 3, 4$. Convergence tolerance was set to $\texttt{tol} = 1\text{e}{-8}$. Solid lines correspond to standard Ritz shifts, dashed lines to harmonic Ritz shifts.

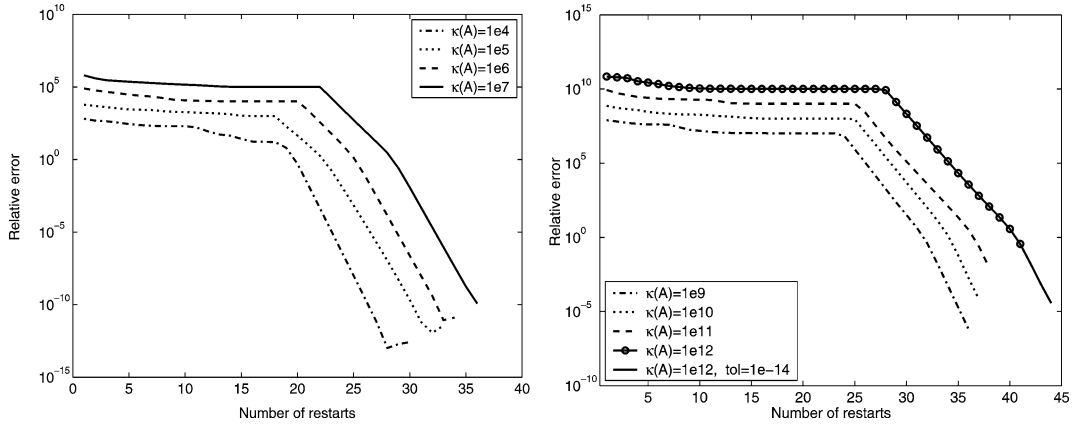


Fig. 2. Experiments with the increasingly ill-conditioned matrices (23), with condition numbers $\kappa(A) = 10^s$, $s = 4, 5, 6, 7$ (left) and $\kappa(A) = 10^s$, $s = 9, 10, 11, 12$ (right).

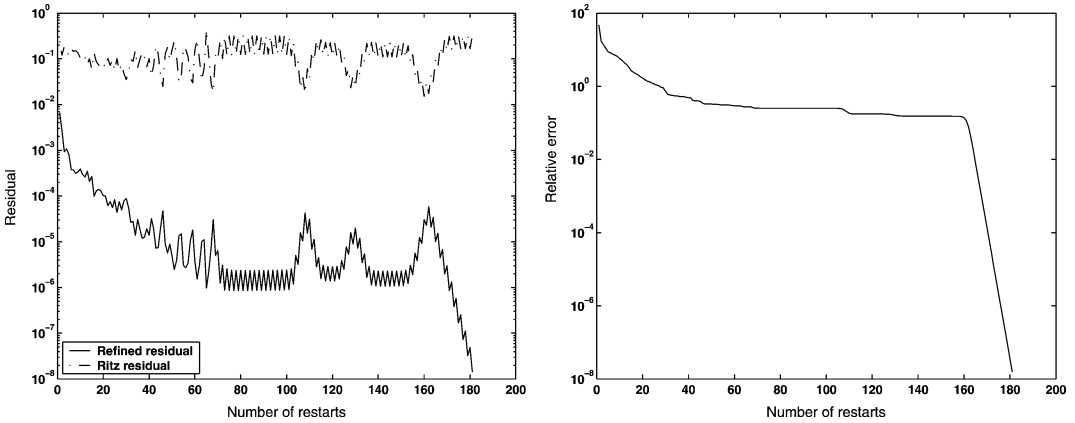


Fig. 3. Experiments with IRLANB for matrix illc1850.

relative error; the left diagram shows residual estimates provided by the refined as well as by the Ritz residuals. Observe that IRLANB converged with relative accuracy $O(10^{-8})$ and that the refined residuals provided an accurate gauge of convergence, in contrast to the Ritz residuals, that failed to detect it.

### 8.3. Computing few singular values

We next illustrate the ability of IRLANB to quickly detect a few additional singular values that lie near the smallest, once the latter has converged. We continue using IRLANB with harmonic Ritz values.

We first experiment with matrix grcar of dimension $N = 1000$ [19] included in MATLAB's function gallery. Our target is to compute its 10 smallest singular values. The length of LBD was $l = k + p = 40$ while we used $p = 10$ implicit shifts per step. Fig. 4 illustrates the norms of the residual for each iteration. The dashed lines represent the convergence criterion that was set equal to $\texttt{normest}(A) \times \texttt{tol}$, where $\texttt{normest}(A)$ is an estimation of the norm of $A$ which we approximate by $\|A\|_2 \approx \|B_l\|_2$ (computed before the first restart). We conducted two experiments. In the first case (top of Fig. 4) we used convergence tolerance equal to $\texttt{tol} = 1\text{e}{-6}$ while in the second case we used $\texttt{tol} = 1\text{e}{-10}$. The plots on the right of Fig. 4 are detailed versions of the plots on the left. We immediately notice that IRLANB can continue computing singular values at subsequent restarts. This behavior is even more pronounced when we employ a stricter convergence tolerance ($\texttt{tol} = 1\text{e}{-10}$). Notice that after the smallest singular value has been approximated (at restart number 98), then in the subsequent 9 restarts each of the remaining singular values is approximated. Observe that the ratio among the largest and smallest singular values computed is $\frac{\sigma_{N-9}}{\sigma_N} = 1.0027$. Obviously, deflation has helped to deal effectively with this level of clustering.

We next experiment with matrix dw_2048 from Matrix Market, the 10 smallest singular values of which are not as clustered as in the previous case ($\frac{\sigma_{N-9}}{\sigma_N} = 21.9$). We used $k + l = 50$, $p = 20$ and experimented with convergence tolerances $\texttt{tol} = 1\text{e}{-8}$ and $\texttt{tol} = 1\text{e}{-12}$. Fig. 5 illustrates the results. As in the experiment with grcar we observe that IRLANB rapidly approximates the remaining singular values once convergence for the smallest one has been achieved. Because of the decreased clustering of the smallest singular values, however, convergence is not as fast as in the previous case.

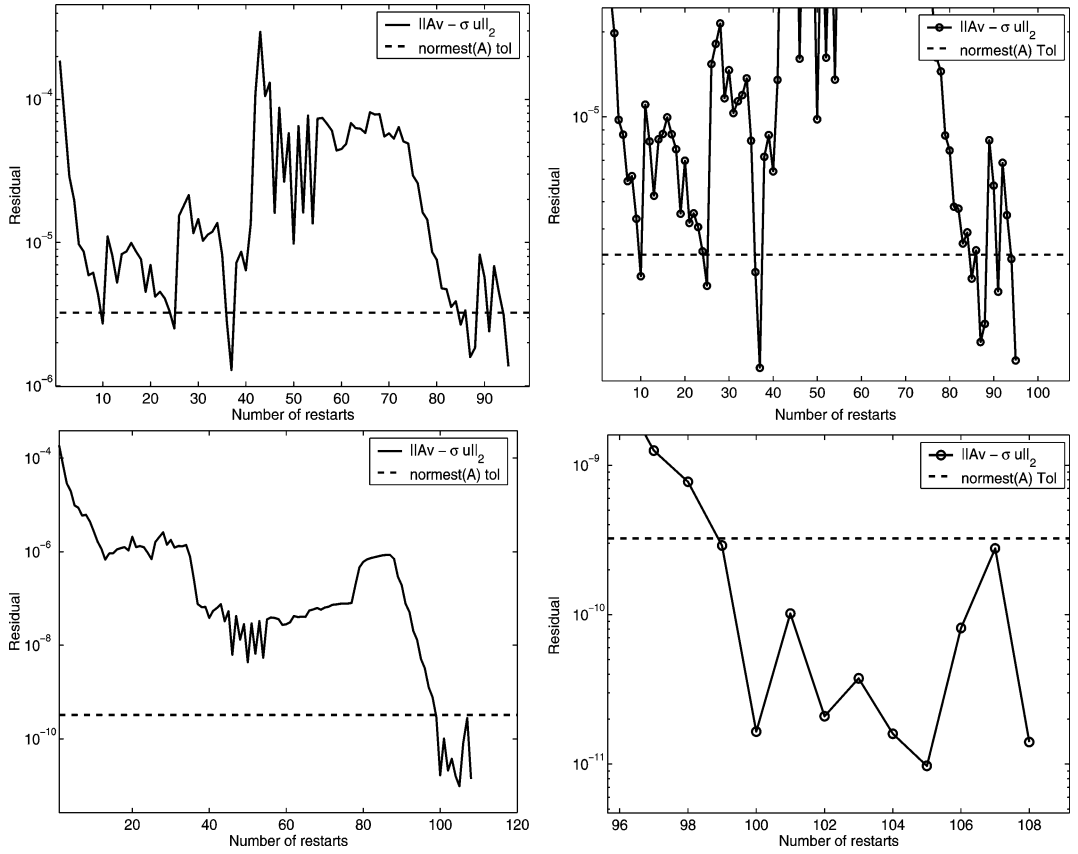


Fig. 4. Experiments with `IRLANB` on `grcar(1000)`. Up: Convergence tolerance `tol` = 1e−6. Down: Convergence tolerance `tol` = 1e−10.

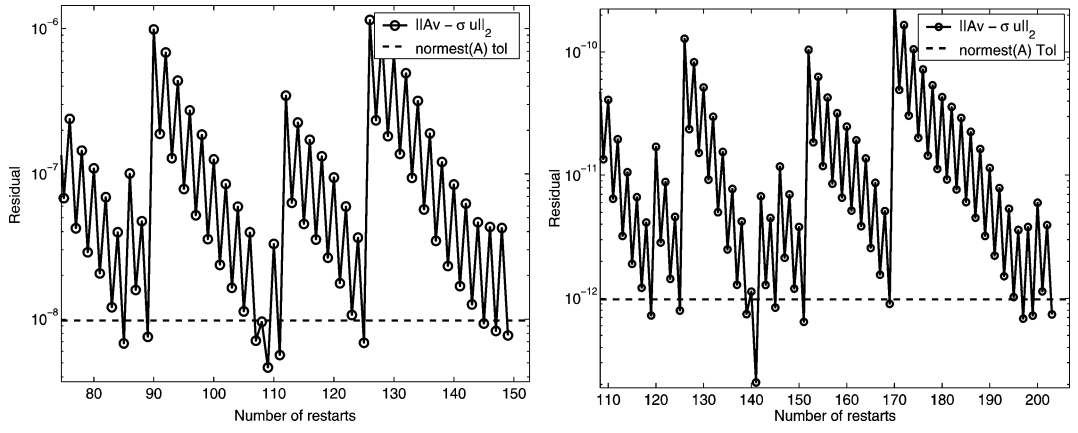


Fig. 5. Experiments with `IRLANB` on `dw_2048`. Left: Convergence tolerance `tol` = 1e−8. Right: Convergence tolerance `tol` = 1e−12.

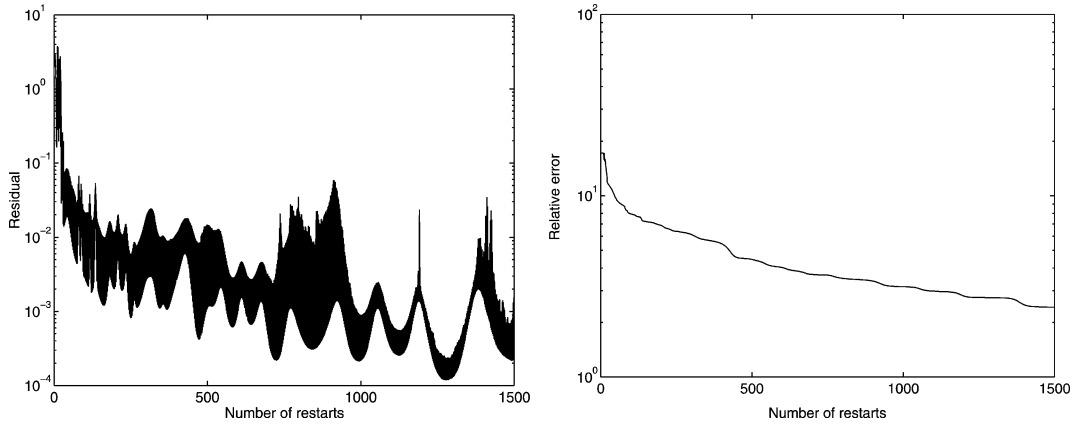


Fig. 6. Experiments with IRLANB for matrix west0479.

We next show an example with highly clustered singular values and high condition ($\kappa(A) = \mathrm{O}(10^{11})$) that causes problems for IRLANB. This is matrix west0479 from Matrix Market and parameters $k + p = 40$, $p = 10$, tol $= 1\mathrm{e}{-12}$ and maxit $= 1500$. Fig. 6 illustrates the disappointing performance of the method, which fails to locate the minimum singular value within a reasonable number of restarts.

### 8.4. Comparisons with related methods

In this section we provide numerical experiments that illustrate the behavior of IRLANB vis-a-vis the methods selected above, namely IRBLEIGS-IRLBSVDS and JDQZ. In both algorithms, the singular values are obtained via the augmented matrix $C$. The first two examples are with two matrices, namely jpwh_991 ($991 \times 991$, nnz $= 6027$) and well_1850 ($1850 \times 712$, nnz $= 8755$), both obtained from Matrix Market. We used the algorithms under consideration to compute one as well as two of the smallest singular triplets. The convergence tolerance was set to tol $= 1\mathrm{e}{-6}$. The minimum search space dimension (k for IRLANB, jmin for JDQZ, and BLSZ for IRBLEIGS-IRBLSVDS) was set to 3. The maximum search space dimensions used were set as k + p = jmax = NBLS × BLSZ = 15; cf. the help pages of IRBLEIGS-IRBLSVDS and JDQZ for detailed explanation regarding the input parameters. Tables 1 and 2 illustrate the corresponding number of restarts and matrix vector products, and indicates that IRLANB competes well with modern available methods. Note that the number of matrix vector products is multiplied by two for IRBLEIGS-IRBLSVDS and JDQZ, since the matrix vector product using the augmented matrix $C$, is equivalent to one matrix vector with $A$ and one with $A^*$.

Our last experiment originates from the computation of pseudospectra of large matrices. Since the $\epsilon$-pseudospectrum of a matrix can be defined as the locus of points $z$ of the complex plane that satisfy the inequality $\sigma_{\min}(zI - A) \leqslant \epsilon$, it becomes of critical importance to use fast algorithms to estimate the smallest singular value (see [49] for a comprehensive survey). We experiment with a family of matrices, studied in [51], that originate from specific bidiagonal ones to which we add random sparse entries. In MATLAB notation, the matrices are defined as

$$\begin{aligned} \mathtt{A} = {} & \mathtt{spdiags}([3 * \exp(-(0:\mathtt{N}-1)'/10), 0.5 * \mathtt{ones}(\mathtt{N},1)], 0:1, \mathtt{N}, \mathtt{N}) \ldots \\ & + 0.1 * \mathtt{sprandn}(\mathtt{N}, \mathtt{N}, 10/\mathtt{N}), \end{aligned} \tag{23}$$

Table 1
Number of restarts (IT), matrix vector products (MV) and runtimes (in sec) for matrix `jpwh_991` in order to compute one or two of the smallest singular triplets

| `jpwh_991` | `IRLANB` | | | `IRBLEIGS-IRBLSVDS` | | | `JDQZ` | | |
|---|---|---|---|---|---|---|---|---|---|
| # min. triplets | IT | MV | sec | IT | MV | sec | IT | MV | sec |
| 1 | 22 | 580 | 3.6 | 210 | 6300 | 18.25 | 294 | 3532 | 35 |
| 2 | 82 | 2080 | 11.6 | 445 | 13350 | 38.25 | 336 | 4036 | 41.4 |

Table 2
Number of restarts (IT), matrix vector products (MV) and runtimes (in sec) for matrix `well_1850` in order to compute one or two of the smallest singular triplets

| `well_1850` | `IRLANB` | | | `IRBLEIGS-IRBLSVDS` | | | `JDQZ` | | |
|---|---|---|---|---|---|---|---|---|---|
| # min. triplets | IT | MV | sec | IT | MV | sec | IT | MV | sec |
| 1 | 106 | 2680 | 21.86 | 230 | 6900 | 39.09 | `FAILED` | | |
| 2 | 118 | 2980 | 25.66 | 268 | 8040 | 47.22 | `FAILED` | | |

Table 3
Number of restarts (IT), matrix vector products (MV), runtimes (in sec) and approximations of $\sigma_{\min}$ for the family of random matrices (23). The star ("$*$") indicates that the method ran out of memory (1 GByte)

$z = 3.5$

| $N \times 10^4$ | `IRLANB` | | | | `IRBLEIGS-IRBLSVDS` | | | | `JDQZ` | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\hat{\sigma}_{\min}$ | IT | MV | sec | $\hat{\sigma}_{\min}$ | IT | MV | sec | $\hat{\sigma}_{\min}$ | IT | MV | sec |
| 5 | 0.3725 | 1 | 91 | 47 | 0.3725 | 5 | 300 | 78 | 0.3725 | 23 | 306 | 273 |
| 10 | 0.3740 | 1 | 91 | 102 | 0.3740 | 7 | 420 | 210 | 0.3740 | 23 | 306 | 436 |
| 15 | 0.3726 | 1 | 91 | 150 | 0.3726 | 7 | 420 | 331 | * | * | * | * |
| 20 | 0.3718 | 1 | 91 | 203 | 0.3718 | 8 | 480 | 505 | * | * | * | * |

$z = 1$

| $N \times 10^4$ | `IRLANB` | | | `IRBLEIGS-IRBLSVDS` | | | `JDQZ` | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\hat{\sigma}_{\min}$ | IT (MV) | sec | $\hat{\sigma}_{\min}$ | IT (MV) | sec | $\hat{\sigma}_{\min}$ | IT (MV) | sec |
| 5 | 1.04e−4 | 7 (271) | 218 | 1.04e−4 | 19 (1140) | 294 | 1.84e−1 | 264 (3194) | 3740 |
| 10 | 5.23e−4 | 6 (241) | 500 | 5.23e−4 | 19 (1140) | 601 | 1.91e−1 | 148 (1802) | 4080 |
| 15 | 0.0014 | 6 (241) | 741 | 0.0014 | 23 (1380) | 1141 | * | ** | * |
| 20 | 8.84e−6 | 7 (271) | 1138 | 8.84e−6 | 19 (1140) | 1270 | * | ** | * |

where `N` is the size of the matrix. Specifically, we seek $\sigma_{\min}(A - zI)$ for values $z = 1$ and $z = 3.5$, and dimensions $\mathtt{N} = 50000 : 50000 : 200000$. The parameters for `IRLANB` and `JDQZ` were: Minimum dimension of search space $\mathtt{k} = \mathtt{jmin} = 15$ and maximum dimension of search space $\mathtt{k} + \mathtt{p} = \mathtt{jmax} = 30$. Convergence tolerance was set to $\mathtt{tol} = 1\mathrm{e}{-10}$. The corresponding parameters for `IRBLEIGS-IRBLSVDS` were $\mathtt{BLSZ} = 3$, $\mathtt{NBLS} = 10$ and $\mathtt{tol} = 1\mathrm{e}{-6}$. The maximum number of restarts was set to $\mathtt{MAXIT} = 1000$. Table 3 illustrates the number of restarts and matrix vector products, as well as convergence results. We observe that for the shift $z = 3.5$, all three methods return similar results (up

to 4 digits); however, IRLANB requires significantly fewer matrix–vector products. Furthermore, when $N = 150000, 200000$ we observe that JDQZ ran out of memory for both shifts. Finally, we note that the $\hat{\sigma}_{\min}$ computed by JDQZ for the shift $z = 1$ is entirely different than the results of the other two methods that are in agreement in 9 to 10 digits.

Selecting the parameters $k$ and $p$ for the implicit restarts does not follow any specific rule, but rather depends upon the computational resources at our disposal. Our experience indicates that, in general, $k$ should be at least twice the number of the singular values sought. If $p$ is close to $k$ then the iteration tends to be more "aggressive", in the sense that it may require fewer restarts to converge to the smallest singular value, but may pose problems if we seek several singular values.

## 9. Conclusions

In this paper we described the design of IRLANB, an implicitly restarted Lanczos bidiagonalization algorithm for the computation of a few of the smallest singular values of a matrix. We investigated Ritz as well as harmonic Ritz values as shifts in the implicit QR steps and demonstrated the superiority of the latter in the case of clustered smallest singular values. We showed how to efficiently compute the harmonic Ritz values, only at a very small additional cost compared to Ritz values. Furthermore, we demonstrated that IRLANB with harmonic Ritz values can successfully compute the smallest singular value of matrices with very large condition numbers. We proved that the orthogonal deflation transformation can be applied directly on Lanczos bidiagonalization. Numerical experiments demonstrate that this deflation scheme can efficiently compute clustered singular values. Finally, we demonstrated the application of refined residuals and vectors in the case of Lanczos bidiagonalization. The computation of the smallest singular values is a difficult and computationally challenging problem. We believe that the above framework will prove to be very helpful in future investigations as well as in practical computations.

## Acknowledgements

We are grateful to one referee for his/her painstaking review and extensive comments that helped us improve the paper considerably. A first version of this work was developed in the context of the first author's diploma thesis [26] and presented during the International Workshop on Parallel Matrix Algorithms and Applications (PMAA '02) held in Neuchâtel. We thank Errikos Kontoghiorghes for his hospitality during that Workshop. We would also like to thank Valeria Simoncini for bringing to our attention [29], PROPACK and suggesting several improvements, especially the compact formulation of Proposition 4.1; Henk van der Vorst for helpful discussions during HERCMA '01 in Athens and his suggestion that we consider using harmonic Ritz values; Gerard Sleijpen for useful comments concerning stability issues; Michiel Hochstenbach for bringing to our attention [25]; and Andreas Stathopoulos for many insightful discussions regarding all aspects of this work. The IRLANB code is available from the authors upon request.

## References

[1] J. Baglama, D. Calvetti, L. Reichel, IRBL: An implicitly restarted block Lanczos method for large-scale Hermitian eigenproblems, SIAM J. Sci. Comput. 24 (5) (2003) 1650–1677.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, PA, 2000.
[3] M.W. Berry, Large scale singular value decomposition, Internat. J. Supercomp. Appl. 6 (1992) 13–49.
[4] M.W. Berry, D. Mezher, B. Philippe, A. Sameh, Parallel computation of the singular value decomposition, Technical Report No. 4694, INRIA, Rennes, January 2003.
[5] Å. Björck, E. Grimme, P. Van Dooren, An implicit bidiagonalization algorithm for ill-posed systems, BIT 34 (1994) 510–534.
[6] Å. Björck, E. Grimme, P. Van Dooren, Methods for large scale total least squares problems, SIAM J. Matrix Anal. Appl. 22 (2000) 413–429.
[7] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, PA, 1996.
[8] T. Braconnier, N.J. Higham, Computing the field of values and pseudospectra using the Lanczos method with continuation, BIT 36 (3) (1996) 422–440.
[9] P. Bradley, O. Mangasarian, $k$-plane clustering, J. Global Optimization (1999) 1–9.
[10] D. Calvetti, L. Reichel, D.C. Sorensen, An implicitly restarted Lanczos method for large symmetric eigenvalue problems, Elec. Trans. Numer. Anal. 2 (1994) 1–21.
[11] J. Cullum, R. Willoughby, M. Lake, A Lanczos algorithm for computing singular values and vectors of large matrices, SIAM J. Sci. Statist. Comput. 4 (2) (1983) 197–215.
[12] J.K. Cullum, R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations, SIAM, Philadelphia, PA, 2002.
[13] L. Elsner, C. He, An algorithm for computing the distance to uncontrollability, Systems Control Lett. 17 (1991) 453–464.
[14] D.R. Fokkema, G.A.G. Sleijpen, H.A. van der Vorst, Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils, SIAM J. Sci. Comput. 20 (1) (1998) 94–125.
[15] G.H. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Numer. Anal. 2 (1965) 205–224.
[16] G.H. Golub, H.A. van der Vorst, Eigenvalue computation in the 20th century, J. Comput. Appl. Math. 123 (2000) 35–65.
[17] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[18] V. Heuveline, B. Philippe, M. Sadkane, Parallel computation of spectral portrait of large matrices by Davidson type methods, Numer. Algorithms 16 (1998) 55–75.
[19] N.J. Higham, The Test Matrix Toolbox for MATLAB (Version 3.0), Technical Report 276, Manchester Centre for Computational Mathematics, September 1995.
[20] M.E. Hochstenbach, A Jacobi–Davidson type SVD method, SIAM J. Sci. Comput. 23 (2) (2001) 606–628.
[21] M.E. Hochstenbach, Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems. Preprint 1263, Department Math., Utrecht University, Utrecht, December 2002.
[22] M.E. Hochstenbach, Subspace Methods for Eigenvalue Problems, Ph.D. Thesis, Dept. Mathematics, Utrecht University, Utrecht, 2003.
[23] H. van der Vorst, Computational methods for large eigenvalue problems, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, vol. 8, North-Holland, Amsterdam, 2002, pp. 3–179.
[24] Z. Jia, Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, Linear Algebra Appl. 259 (1997) 1–23.
[25] Z. Jia, D. Niu, An implicitly restarted refined bidiagonalization Lanczos method for computing a partial singular value decomposition, SIAM J. Matrix Anal. Appl. 25 (1) (2003) 246–265.
[26] E. Kokiopoulou, Parallel Iterative Methods for the Computation of the Smallest Singular Values of Large Sparse Matrices and Applications, Master's Thesis, Comp. Eng. & Informatics Dept., University of Patras, Patras, 2002.
[27] Y.L. Lai, K.Y. Lin, W.W. Lin, An inexact inverse iteration for large sparse eigenvalue problems, Numer. Linear Algebra Appl. 4 (1997) 425–437.
[28] R.M. Larsen, PROPACK: A software package for the symmetric eigenvalue problem and singular value problems on Lanczos and Lanczos bidiagonalization with partial reorthogonalization, SCCM, Stanford University http://sun.stanford.edu/~rmunk/PROPACK/.
[29] R.M. Larsen, Efficient Algorithms for Helioseismic Inversion, Ph.D. Thesis, Dept. Computer Science, University of Aarhus, Denmark, October 1998.
[30] R.M. Larsen, Lanczos bidiagonalization with partial reorthogonalization, Technical Report DAIMI PB-357, Aarhus University, September 1998.

[31] R.M. Larsen, Combining implicit restarts and partial reorthogonalization in Lanczos bidiagonalization. SCCM, Stanford University, April 2001, http://sun.stanford.edu/~rmunk/PROPACK/talk.rev3.ps.gz.
[32] R. Lehoucq, D.C. Sorensen, C. Yang, Arpack User's Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods, SIAM, Philadelphia, PA, 1998, http://www.caam.rice.edu/software/ARPACK.
[33] R.B. Lehoucq, D.C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, SIAM J. Matrix Anal. Appl. 17 (1996) 789–821.
[34] D. Mezher, B. Philippe, Parallel computation of the pseudospectrum of large matrices, Parallel Comput. 28 (2) (2002) 199–221.
[35] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1112–1135.
[36] C.C. Paige, The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices, Ph.D. Thesis, London University, London, 1971.
[37] C.C. Paige, M.A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1982) 43–71.
[38] C.C. Paige, B.N. Parlett, H.A. van der Vorst, Approximate solutions and eigenvalue bounds from Krylov subspaces, Numer. Linear Algebra Appl. 2 (2) (1995) 115–134.
[39] B.N. Parlett, The Symmetric Eigenvalue Problem, SIAM, Philadelphia, PA, 1998.
[40] A. Ruhe, Rational Krylov algorithms for nonsymmetric eigenvalue problems II. Matrix pairs, Linear Algebra Appl. 197/198 (1994) 283–295.
[41] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Halstead Press, New York, 1992.
[42] A. Sameh, Z. Tong, The trace minimization method for the symmetric generalized eigenvalue problem, J. Comput. Appl. Math. 123 (2000) 155–170.
[43] H.D. Simon, The Lanczos algorithm with partial reorthogonalization, Math. Comput. 42 (165) (1984) 115–142.
[44] H.D. Simon, H. Zha, Low-rank matrix approximation using the Lanczos bidiagonalization process with applications, SIAM J. Sci. Comput. 21 (6) (2000) 2257–2274.
[45] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM Rev. 42 (2) (2000) 267–293.
[46] D. Sorensen, Deflation for implicitly restarted Arnoldi methods, Technical Report 98-12, Rice University, CAAM Technical Report, June 1998 (rev. July 2000).
[47] D.C. Sorensen, Implicit application of polynomial filters in a $k$-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1992) 357–385.
[48] G.W. Stewart, Matrix Algorithms, vol. II: Eigensystems, SIAM, Philadelphia, PA, 2001.
[49] L.N. Trefethen, Computation of pseudospectra, Acta Numer. 8 (1999) 247–295.
[50] A.-J. van der Veen, E.F. Deprettere, A.L. Swindlehurst, Subspace based signal analysis using singular value decomposition, Proc. IEEE 8 (1993) 1277–1309.
[51] T. Wright, L.N. Trefethen, Large-scale computation of pseudospectra using ARPACK and Eigs, SIAM J. Sci. Comput. 23 (2) (2001) 591–605.
[52] K. Wu, H. Simon, A parallel Lanczos method for symmetric generalized eigenvalue problems, Technical Report 41284, Lawrence Berkeley National Laboratory, 1997, http://www.nersc.gov/research/SIMON/planso.html.